EgoAgent: A Joint Predictive Agent Model in Egocentric Worlds

Supplementary Material

1. Additional Results on Egocentric Future State Prediction

In this section, we provide additional qualitative results on the egocentric future state prediction task. Additionally, we describe our approach to fine-tune a video diffusion model, OpenSora [9], on the Ego-Exo4D dataset [2] and generate future video frames conditioned on initial frames as shown in Fig. 1.

1.1. Visualizations and Comparisons

We provide more visualizations of the prediction results from our EgoAgent, DoRA [6], and fine-tuned OpenSora [9] in different scenes in Fig. 2. For OpenSora, when predicting the states of t_k, we use all the ground truth frames from t_0 to t_{k-1} as conditions. As OpenSora takes only past observations as input and neglects human motion, it performs well only when the human has relatively small motions (see the top two cases in Fig. 2), but can not adjust to large movements of the human body or quick viewpoint changes (see the bottom two cases in Fig. 2).

1.2. Finetuning OpenSora on Ego-Exo4D

OpenSora [9], initially trained on Internet videos and images, produces severely inconsistent results when directly applied to infer future videos on the Ego-Exo4D dataset, as illustrated in Fig. 1. To address the gap between general Internet content and egocentric video data, we fine-tuned the official OpenSora V1.1 checkpoint on the Ego-Exo4D training set for 50 epochs. OpenSora V1.1 proposed a random mask strategy during training to enable video generation by image and video conditioning. We adopted the default masking rate, which applies: 75% with no masking, 2.5% with random masking of 1 frame to 1/4 of the total frames, 2.5% with masking at either the beginning or the end for 1 frame to 1/4 of the total frames, and 5% with random masking spanning 1 frame to 1/4 of the total frames at both the beginning and the end.

Figure 1. Comparison of OpenSora V1.1 first-frame-conditioned video generation results before and after finetuning on Ego-Exo4D. Fine-tuning enhances temporal consistency, but the predicted pixel-space future states still exhibit errors, such as inaccuracies in the basketball's trajectory.

As shown in Fig. 1, despite being trained on a large dataset, OpenSora struggles to generalize to the Ego-Exo4D dataset, producing future video frames with minimal consistency relative to the conditioning frame. While fine-tuning improves temporal consistency, the moving trajectories of objects like the basketball and soccer ball still deviate from real physical laws. Compared with our feature space prediction results, this suggests that training world models in a reconstructive latent space is more challenging than training them in a feature space.

2. Additional Results on 3D Human Motion Prediction

We present additional qualitative results for the 3D human motion prediction task, highlighting a particularly challenging scenario where egocentric observations exhibit minimal variation. This scenario poses significant difficulties for video-conditioned motion prediction, as the model must effectively capture and interpret subtle changes. As demonstrated in Fig. 3, EgoAgent successfully generates accurate predictions that closely align with the ground truth motion, showcasing its ability to handle fine-grained temporal dynamics and nuanced contextual cues.

3. Action Affects the Perceptual Area

We visualize the attention map of EgoAgent's first transformer head in the representation and action prediction tasks. As shown in Fig. 4, when provided with only the egocentric image, EgoAgent focuses on the ball. Upon receiving the action query token, it shifts attention to the human body part (the right foot at t_0 and the left foot at t_1) in line with the ground truth body motion, indicating that actions can guide the model to focus on task-related areas.

4. OpenSora for Image Classification

In this section, we detail the process of extracting features from OpenSora V1.1 [9] (without fine-tuning) for an im-

Figure 2. Retrieval and generation results for egocentric future state prediction. Correct and wrong retrieval images are marked with green and red borders, respectively.

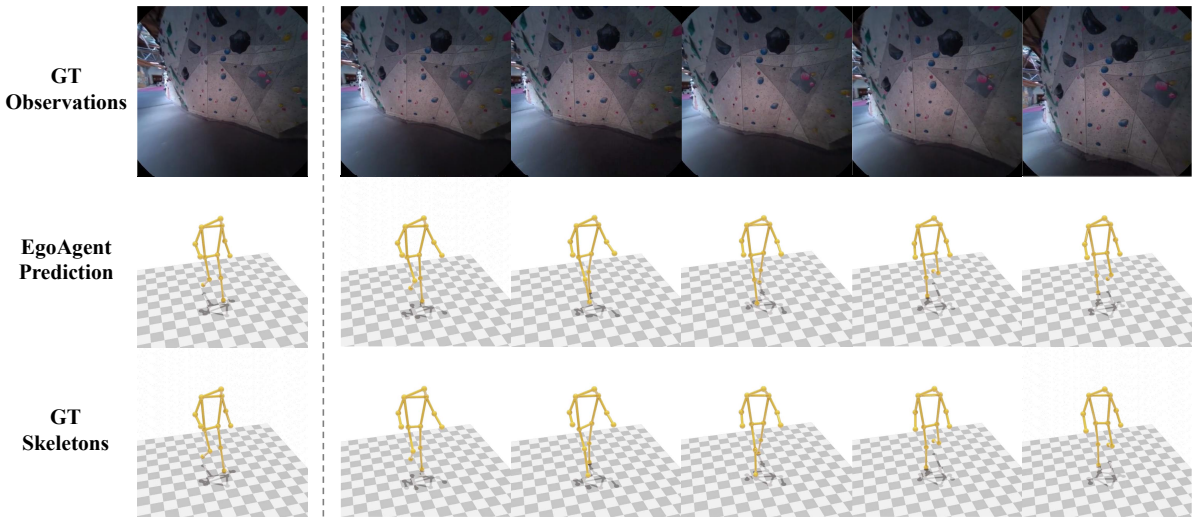

Figure 3. 3D Human motion prediction results in scenes with minor changes in egocentric observations.

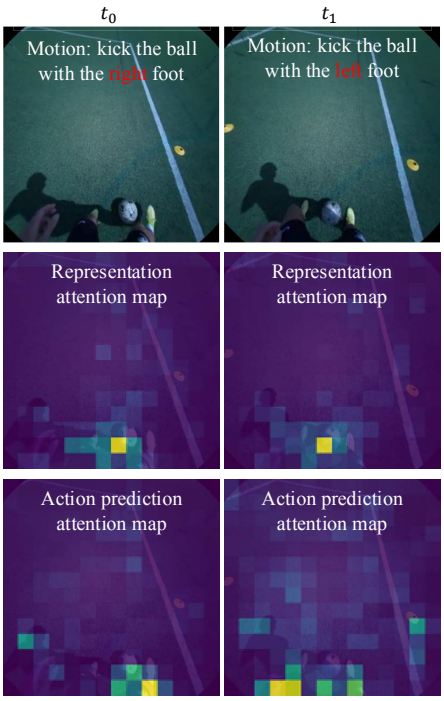

Figure 4. Attention map of EgoAgent performing visual representation and action prediction task. EgoAgent attends to the moving body part in the image when predicting future actions.

age classification task. Following the approach of [7], we leverage the insight that diffusion models can be interpreted as multi-level denoising autoencoders. These models inherently learn linearly separable representations within their intermediate layers, without relying on auxiliary encoders. The quality of the extracted features depends on both the layer depth and the noise level applied during extraction.

As shown in Tab. 1, we first evaluate k-NN classification performance on the ImageNet-100 dataset using three intermediate layers and five different noise scales. We find that a noise timestep of 128 yields the best results, with the middle and last layers performing significantly better than the first layer. We then test this optimal configuration on ImageNet-1K and find that the last layer with 128 noising timesteps achieves the best classification accuracy.

Table 1. k-NN evaluation results of OpenSora V1.1 features from different layer depths and noising scales on ImageNet-100. Top1 and Top5 accuracy (%) are reported.

Timesteps	First Layer		Middle Layer		Last Layer	
	Top1	Top5	Top1	Top5	Top1	Top5
32	6.10	18.20	34.04	59.50	30.40	55.74
64	6.12	18.48	36.04	61.84	31.80	57.06
128	5.84	18.14	38.08	64.16	33.44	58.42
256	5.60	16.58	30.34	56.38	28.14	52.32
512	3.66	11.70	6.24	17.62	7.24	19.44

5. Data Preprocess

For egocentric video sequences, we utilize videos from the Ego-Exo4D [2] and WalkingTours (WT) [6] datasets. The original resolution of Ego-Exo4D videos is 1408×1408, captured at 30 fps. We sample one frame every five frames and use the original resolution to crop local views (224×224) for computing the self-supervised representation loss [1]. For computing the prediction and action loss, the videos are downsampled to 224×224 resolution. WT primarily consists of 4K videos (3840×2160) recorded at 60 or 30 fps. Similar to Ego-Exo4D, we use the original resolution and downsample the frame rate to 6 fps for representation loss computation. As Ego-Exo4D employs fish-

Table 2. Architecture configurations of EgoAgent.

	EgoAgent-300M	EgoAgent-1B
Depth	22	22
Embedding dim	1024	2048
Number of heads	8	16
MLP ratio	8/3	8/3
#Param.	284M	1.13B

eye cameras, we undistort the images to a pinhole camera model using the official *Project Aria Tools* to align them with the WT videos.

For motion sequences, the Ego-Exo4D dataset provides synchronized 3D motion annotations and camera extrinsic parameters for various tasks and scenes. While some annotations are manually labeled, others are automatically generated using 3D motion estimation algorithms from multiple exocentric views. To maximize data utility and maintain high-quality annotations, manual labels are prioritized wherever available, and automated annotations are used only when manual labels are absent. Each pose is converted into the egocentric camera's coordinate system using transformation matrix derived from the camera extrinsics. These transformation matrices also enable the computation of trajectory vectors for each frame in a sequence. Beyond the x, y, z coordinates, a visibility dimension is appended to account for keypoints invisible to all exocentric views. Finally, a sliding window approach segments sequences into fixed-size windows to serve as input to the model. Note that we do not downsample the frame rate of 3D motions.

6. Training Details

6.1. Architecture Configurations

In Tab. 2, we provide detailed architecture configurations for EgoAgent following the scaling-up strategy of InternLM [5]. To maintain the scaling-up and generalization ability, we do not modify the internal modules in InternLM, *i.e.*, we adopt the RMSNorm [8] and 1D RoPE [4]. We show that, without specific modules designed for vision tasks, EgoAgent can perform well on egocentric vision and action tasks.

Tab. 3 presents the detailed configuration of the embedding and prediction modules in EgoAgent, including the image projector (Proj_i), representation head/state prediction head (MLP_i), action projector (Proj_a) and action prediction head (MLP_a). Note that the representation head and the state prediction head share the same architecture but have distinct weights.

6.2. Training Configurations

In Tab. 4, we provide the detailed training hyper-parameters for experiments in the main manuscripts. The training uses the AdamW optimizer [3] with momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The base learning rate is set at 6×10^{-4}, with a cosine learning rate schedule and a base weight decay of 0.04, transitioning to an end weight decay of 0.4. A batch size of 1920 is employed for $72,000$ iterations, with $1,800$ warm-up iterations using a linear schedule. Gradient clipping is applied at a value of 1.0, and data is processed in Float-16 precision. Additionally, the exponential moving average (EMA) momentum is set to 0.996. The normalization epsilon is fixed at 1×10^{-6} for stability in training.

Table 3. Architecture of the embedding (Proj_i, Proj_a) and prediction (MLP_i, MLP_a) modules in EgoAgent. For details on module connections and functions, please refer to Fig. 2 in the main paper.

	Norm & Activation	Output Shape
Proj_i (*Image projector*)		
Input image	-	3×224×224
Conv 2D (16×16)	-	Embedding dim×14×14
MLP_i (*State prediction head & Representation head)*		
Input embedding	-	Embedding dim
Linear	GELU	2048
Linear	GELU	2048
Linear	-	256
Linear	-	65536
Proj_a (*Action projector*)		
Input pose sequence	-	4×5×17
Conv 2D (5×17)	LN, GELU	Embedding dim×1×1
MLP_a (*Action prediction head*)		
Input embedding	-	Embedding dim×1×1
Linear	-	4×5×17

Table 4. Hyper-parameters for training EgoAgent.

Training Configuration	EgoAgent-300M/1B
Training recipe:	
optimizer	AdamW [3]
optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.999$
Learning hyper-parameters:	
base learning rate	6.0E-04
learning rate schedule	cosine
base weight decay	0.04
end weight decay	0.4
batch size	1920
training iterations	72,000
learning rate warm-up iterations	1,800
warm-up schedule	linear
gradient clip	1.0
data type	Float-16
norm epsilon	1.0E-06
EMA hyper-parameters:	
momentum	0.996

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 3

[2] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-Exo4D: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024. 1, 3

[3] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[4] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 4

[5] InternLM Team. InternLM: A multilingual language model with progressively enhanced capabilities, 2023. 4

[6] Shashanka Venkataramanan, Mamshad Nayeem Rizve, João Carreira, Yuki M Asano, and Yannis Avrithis. Is ImageNet worth 1 video? Learning strong image encoders from 1 long unlabelled video. *The Twelfth International Conference on Learning Representations*, 2024. 1, 3

[7] Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15802–15812, 2023. 3

[8] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019. 4

[9] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-Sora: Democratizing efficient video production for all, 2024. 1